

TRANSMISSION OF MESSAGE WITH DIGITAL SIGNATURE ON TOUCH SCREEN USING ARM9

V. V. N. S. SUDHA¹ & R. R DHRUVA²

¹Assistant Professor, Department of ECE, Aurora's Technological & Research Institute, Hyderabad, Andhra Pradesh, India

²Senior Assistant Professor, Department of ECE, CVR College of Engineering, Hyderabad, Andhra Pradesh, India

ABSTRACT

This paper focuses on the design and implementation of “**Transmission of Message with Digital Signature on Touch Screen using ARM9**”. The major application of this system is that it can be used either as a paperless fax machine or as a message transmission system. Nowadays fax machines are one of the most important office equipment. The fax machine usually sends and receives a document through a telephone line. Using fax machine (facsimile machine) we can send documents to any point of the world as very convenient and easy. But for sending the data to the other fax machine through it requires paper to send and receive the data. To send the data first we need to scan the paper and then we send later at receiving side they receive the data to view the fax and they modify the data and again the fax to them by using one more paper. So this existing method is using more paper. The important feature of designing paperless fax machine equipped with a single-touch panel is to replace the scanning and the printing units which not only takes up most space but also is costly and vulnerable parts. The proposed system enables Digital Signature, if required, on the touch screen of ARM9 board using Stylus to overcome the drawbacks of the existing method. The user can do the digital Signature, if required, on the touch screen of the board and a copy of text is stored in the pen drive for future references. This is implemented in ARM9 using Mini S3C2440. The OS used is Linux. CPP is the Programming language used along with Qtopia. The message or fax is transferred between two ARM9 boards. The two boards are connected using Ethernet cable.

KEYWORDS: Digital Signature, Qtopia, ARM9

INTRODUCTION

The Transmission of Message with Digital Signature on Touch Screen using ARM9 application is implemented using two ARM9 boards and Qtopia is the cross-compiler used. The block diagram of the application is shown in figure 1. The block diagram consists of two S3C2440 ARM9 Boards connected with Ethernet cable. The transmission of data between the two boards is done through this Ethernet cable. The message to be sent is typed on one of the boards using the keypad and is sent to the other board. The moment the other board gets the message an indication appears on the touch screen within a textbox. The message is seen on the touch screen of other board by clicking retrieve option. If needed, the user at the other end can do the Digital Signature and send the document back. The user can do the signature on touch screen of the board, by selecting the signature option created on the touch-screen of ARM9 board and should click on the save option. A copy of message is saved in the pen-drive for future references. Touch panel acts as an input device as well as displaying device, which can detect the location being touched. Handwriting reconstruction algorithm^[12] is used internally in the board itself to turn a user's signing act into a digital signature close to the original as much as possible. The document containing the message as well as the signature is sent back to the first board and is seen by clicking the retrieve option. Single touch panel added to conventional fax machine-replaces the scanning and printing units. It supports full-duplex communication.

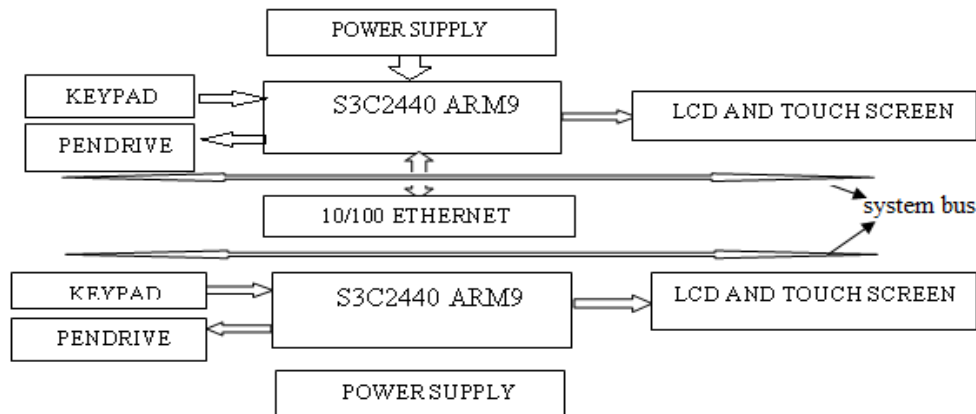


Figure 1: Block Diagram of Transmission of Message with Digital Signature on Touch Screen Using ARM9

Qt (QT) is used to create various components on the touch screen such as widgets. The operating System used is Linux and programming is done using CPP.

The procedure of implementation of this project is as follows:

Step 1: Porting of bootloader, kernel and root file system into the two S3C2440 boards.

Step 2: Developing the application in QT using CPP.

Step 3: Observing the output by transmitting the message between the two ARM9 Boards. Then the message is displayed on the Touch Screens of ARM9 boards. Digital Signature is done using Stylus and a copy of message is stored in the pen drive.

Organization of the work is as follows: In Section 1, we introduce the design of the system hardware, In section 2 mentions the software. In section 3 describes the Experimental Results. Section 4 gives the Conclusion and section 5 discusses the Future scope.

HARDWARE AND SOFTWARE DESIGN OF THE SYSTEM

Hardware Architecture

The system is designed by using two ARM9 (S3C2440) boards which communicate using Ethernet. Touch Screen which is of 3.5 inches is used as display module. Pen drives are attached to the board for the purpose of storage of messages or fax for the future reference. USB is used in the porting procedure.

Samsung S3C2440 Processor

Samsung's S3C2440A^{[1-3][4]} is a 16/32-bit RISC microprocessor. Samsungs S3C2440A is designed to provide hand-held devices and general applications with low-power, and high-performance micro-controller solution in small die size. The ARM processor is a Reduced Instruction Set Computer (RISC). The S3C2440 is a 32 bit microcontroller that internally integrates ARM920T of the ARM Company. ARM920T implements 5-stage pipeline architecture and separate 16KB Instruction cache and 16KB Data cache which are used for faster performance. The S3C2440 have some integrated on-chip functions such as LCD controller, RAM controller, 3 paths UART, 4 paths DMA, 4 path with PWM of Timer, parallel I/O port, 8 channels of 10-bit ADC, the interface of touch screen, I2C interface, two USB interface controllers, two channels SPI, the main frequency of S3C2440 up to 400MHz. Figure 2 show the MINI2440 on-board peripherals layout. In this project the touch screen of S3C2440 is used for the display. The touch screen used in this project is of 3.5 inches. On this touch screen various widgets are created and is used for doing the digital signature. Four-wire resistive touch panel

is used which works on the principal of voltage drop to detect the coordinate on the panel. The touch Screen uses circle extension method for the reconstruction of digital signature on the touch screen.

SOFTWARE ARCHITECTURE

The software used in the proposed system is Qtopia^[10]. The programming language used in Embedded CPP.Qt is used for set of applications with GUI running on desktop environment and general application based on interface. It supports cross compilation for all multiple boards (open source GUI environment). Qtopia controls embedded related devices on target board and flexible to Fedora 9 and ubuntu.

Qt_Embedded is a C++ toolkit for GUI and application development for embedded devices. It runs on a variety of processors, usually with Embedded Linux.Qt_Embedded based applications write to the Linux frame buffer directly and includes several tools to speed and ease development in testing and debugging of applications. Applications targeted for Embedded Linux can be developed using standard Qt tools, including Qt Designer for visual form design, and with tools specifically tailored to the embedded environment.

Firstly, the linux kernel^[3-4] is ported into ARM9 board. Supervivi Bootloader, kernel and root file system are ported. The main functionality of boot loader is to initialize all the devices that are present on the mother board of MINI 2440 and at the same time to find out whether any problem or any other fault is there in the devices that are present on that mother board of MINI 2440. Kernel is the core part of an operating system. Operating system will perform its functionalities like File management, Process management, Memory management, Network management and Interrupt management with the help of the kernel only.

Developing Application in Qtopia

Here are the steps of creating a GUI application for Friendly arm board. You should have development tools ARM-Linux GCC 4.4.3 and ARM-Qtopia 4.6.3 already installed. The following are the steps

- Open Qt designer which is available with "ARM-Qtopia 4.6.3".
- With Qt Designer create the main form.
- Create a script file "build" for building the application.
- Create a script file "formuic" for compiling the .ui form file.
- Run the script
- Create the required files.
- Generate the .pro (project) file using the qmake tool.
- A project file serial.pro has been created.
- Open it with the text editor.
- Build the application.

The following procedure step describes how to upload these files onto our development board. **Qtopia file system**-Create the serial Desktop file, it Takes the binary file and then open hyper terminal using serial cable connect it to the COM0 of mini2440 and COM port of the PC to change the attributes of the uploaded file. It should become executable in order to run our application. `[root@FriendlyARM /]# cd /opt/Qtopia/bin/ [root@FriendlyARM bin]# chmod 777 serial`

Application Procedure

Various widgets namely save, send, retrieve, clear and signature are created in Qtopia which apper on the Touch Screens of the boards. In order to transfer the data between two ARM 9 boards, TCP/IP protocol is used. Here 4 sockets are created. For each board 2 sockets are used. One socket is used for sending data and the other is used to receive data. Two separate ports are used for text and image. Text is nothing but the message which is to be transferred or received.

The various files created in this application are:

- Main.cpp
- MainWindow.cpp
- MainWindow.h
- Signature.h
- Signature.cpp

Main.cpp

This is used to run the main application by initializing the QT Application.

MainWindow.cpp

In this file, 4 sockets are created and function for accepting connection is written. The various functions called in this file are:

On_retrieve_clicked() : This function is used to retrieve data and signature on the touch screen of ARM9 board.

On_save_clicked() : This function is used to save the text in the pen-drive for future references.

On_ send_clicked() : Here, a function checkstate() is called. If checkState returns 0, only text is sent,else both both signature and text is sent.

On_clearclicked() : Text editor gets cleared .

On_Signature_clicked() : The function sig.show() and is called. The window for signature is opened , signature is done and the widget is closed.

On_textEdit_selectionchanged() : Here a function showFullScreen() is called which opens the keypad and enters the message typed.

acceptConnectionc() : This function is used to read the text which is sent by another host.

acceptConnections() : Signature or image is read from another host.

StartReadc() : This function reads the data from the network and stores the content into a file named data.txt. This function also includes the code for textbox which indicates “Message Received” when the board receives the message sent by another board.

StartReads() : This is used to store the image by giving the port number of the image.

Startc (QString address, quint16 port) : It connects to host for the data transfer. Similarly, Starts() is used to connect to the host for the image transfer.

StartTransferc(): It is used to send data from one board to the another.

StartTransfers(): It is used to transfer image from one board to the another.

Keypad.cpp

The keypad is created to enter the text typed by the user on the touch-screen. After the user enters the text, ok option is to be pressed by the user indicating that the text to be sent is typed and the message appears on the touch-screen. A function called buzzerBeep() is called, which gives a beep sound when a key is pressed.

Signature.cpp

In this file, MouseMoveEvent(), MouseReleaseEvent(), setPen() functions are called. MouseMoveEvent() is used to move the Stylus forward when the user does the signature act. setPen() event sets the color of brush as black.set. MouseRelease event clears the widget provided for doing the signature.

Dialogue.cpp

It is used to show the display on window.

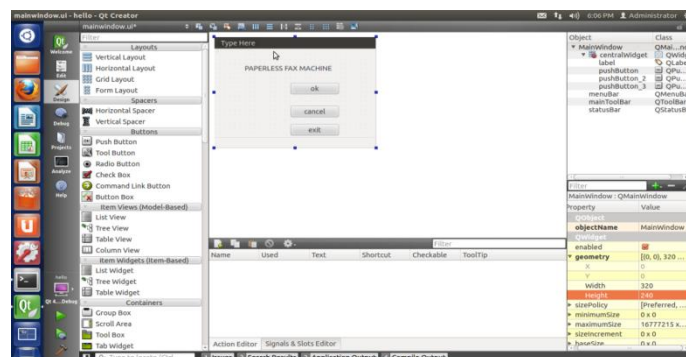


Figure 2: Push Button Creation in QT GUI in Mainwindow.cpp

The above figure illustrates the creation of a push button in Qtopia.

EXPERIMENTAL RESULTS

Figure 3 illustrates the arrangement of two ARM9 boards connected using Ethernet and pendrives attached to each of the boards.



Figure 3: Illustration of Two ARM9 Boards

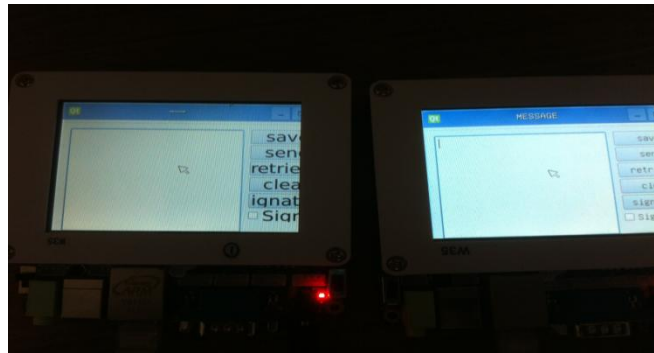


Figure 4: Design of Touch-Screen Using QT

The message is typed in one of the boards using the keypad as shown in figure 5. The option 'Ok' is clicked after typing.

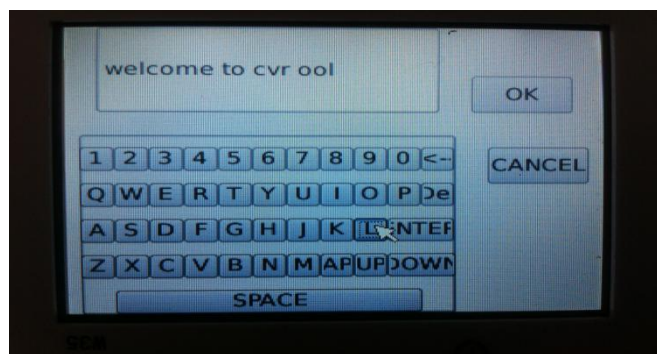


Figure 5: Illustration of Keypad

After clicking 'Ok' option the message appears on the first board as shown in the Figure 6

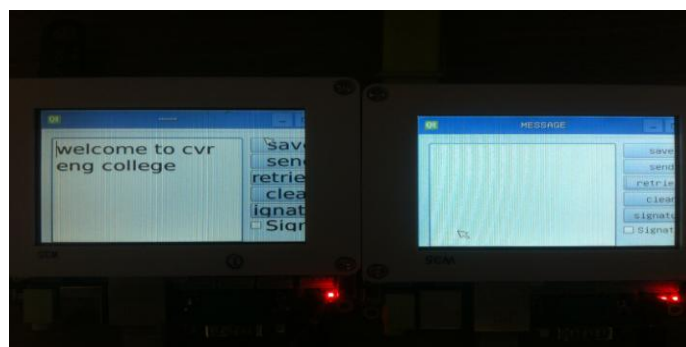


Figure 6: Illustration of the Message Being Sent

The option save is clicked so that the content is saved in the pen-drive as shown in Figure 7.

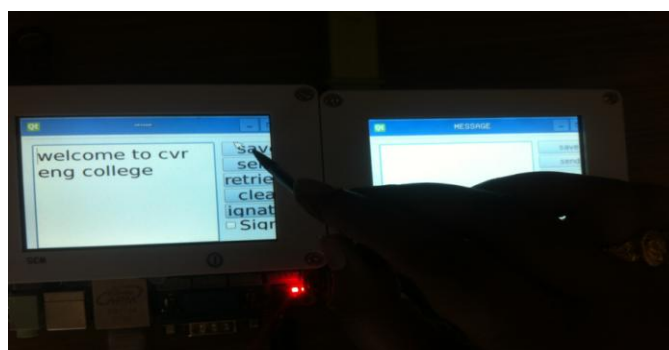


Figure 7: Illustration +of Clicking of Save Option

Then the option 'send' is clicked so that the text is sent to other ARM9 board. On the other board we get the indication as 'Message Received' as shown in Figure 8. Click on ok button.

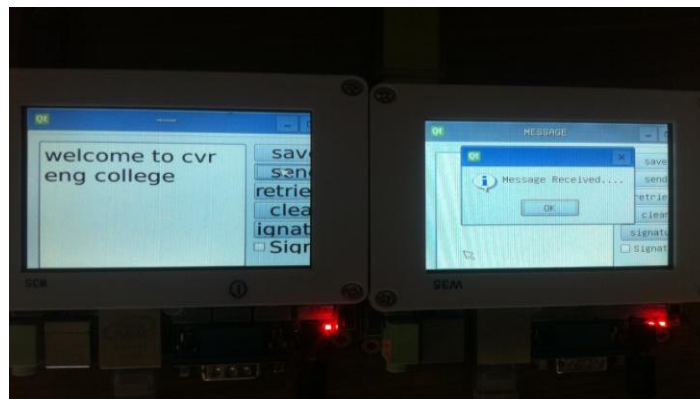


Figure 8: Illustration of Message Received Textbox on the Other Board

The text can be retrieved on the second board as follows by clicking the retrieve option. The message retrieved is shown in Figure 9:

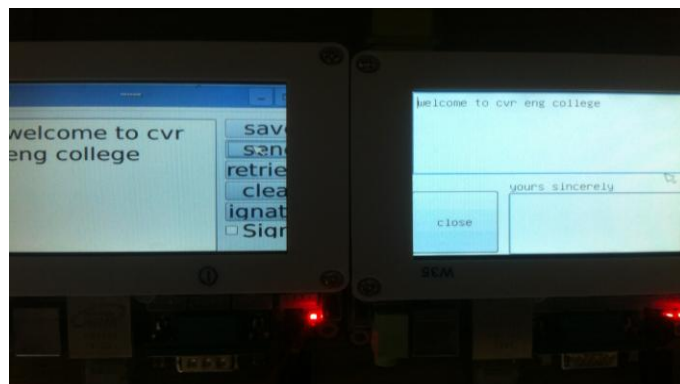


Figure 9: Illustration of the Retrieved Message on the Second Board

If the user needs to do the Digital Signature, the Sign option is to be selected as shown below in Figure 10

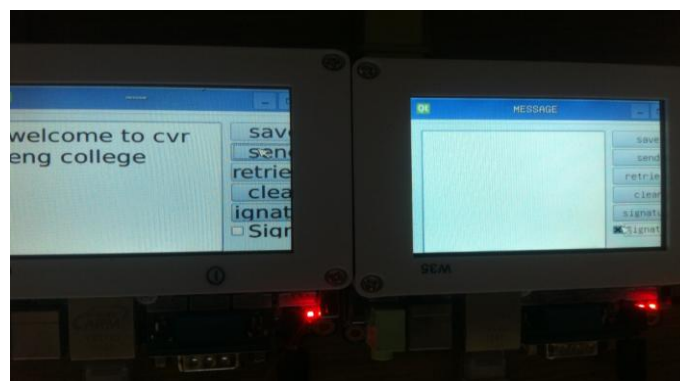


Figure 10: Selection of Signature Option in Order to do the Sign

The Digital Signature is performed in the Widget provided by clicking Signature as shown in Figure 11

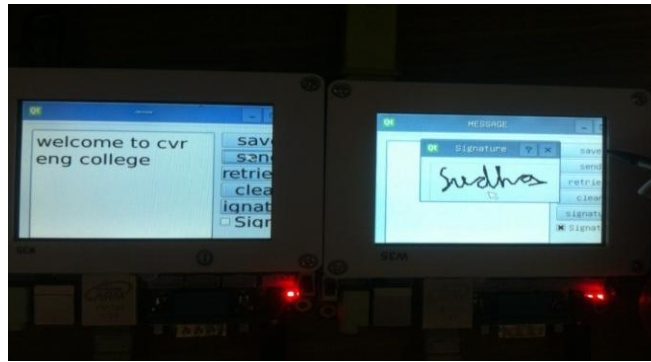


Figure 11: Performing Digital Signature on Touch-Screen

The text as well as the message is sent back to the first board by clicking on the send option, and is retrieved on the first board as shown in Figure 12.

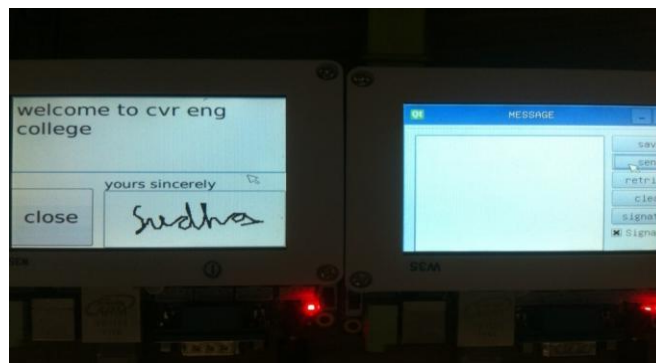


Figure 12: Retrieval of Message and Signature in the First Board as Shown in Figure

The Pen drive can be connected to the system and can be seen as shown in Figure 13

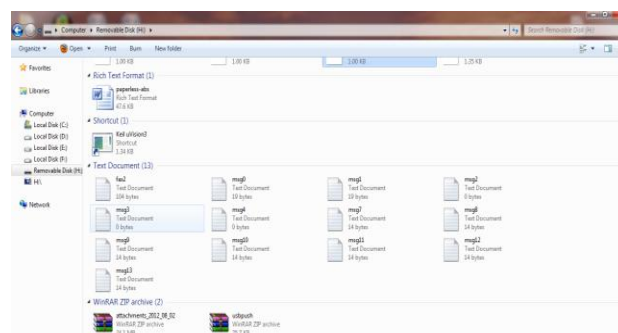


Figure 13: Illustration of Pen Drive Storage

The content of the message stored in pen drive can be read by opening the notepad as shown in Figure 14.

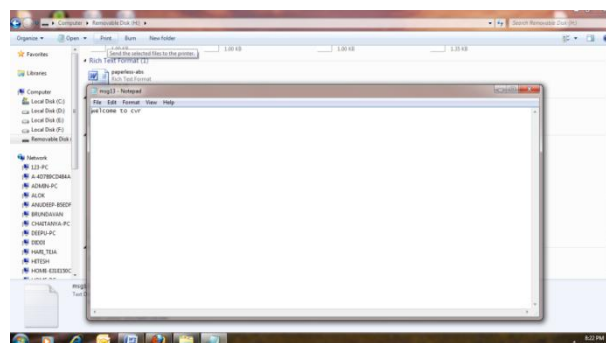


Figure 14: Content of Message Stored in the Pen Drive

CONCLUSIONS

The paper “Transmission of Message with Digital Signature on Touch Screen using ARM9” has been successfully designed, implemented and tested. It has been developed by integrating features of all the hardware components and software used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced ARM9 board and with the help of growing technology the project has been successfully implemented.

FUTURE SCOPE

In the future scope we can go with the usage of touch screen monitor PC's where the ARM board can be directly interfaced to the PC's. Here in this we are going to reduce the extra touch screen panel connected to the ARM board by using touch screen monitors where we can read the received data and we can do the necessary modifications require and can send the modified data and keep the copy of the data inside the memory.

We can also have alert message to your mobiles whenever a new fax is received and also can have alert message with voice by using voice play back recorder circuit like “You received a new FAX”. In this way we are reducing the cost and complexity of the circuit. Here in this concept we are totally eradicating the usage of paper which reduces the environmental pollution.

The concepts of broadcasting and multicasting can be implemented by interfacing more than two ARM9 boards. Using this there will be a possibility of implementation of a distributed system.

WIFI technology can be established for the communication between the two boards.

REFERENCES

1. Sun Yanpeng, peng peng, Zhang Yuan, “Linux transplantation based on the processor S3C2440” 2009.
2. Neil Matthew and Richard Stones , “Beginning Linux Programming”, Third Edition
3. <http://www.friendlyarm.net>
4. www.kernel.org
5. www.linuxjournal.com/magazine
6. <http://www.arm.com/>
7. <http://www.w3schools.com/php>
8. <http://www.datasheetcatalog.com>
9. http://en.wikibooks.org/wiki/C++_Programming
10. <http://www.qtopia.com/>
11. ITU-T.37: Procedures for the Transfer of Facsimile Data via Store-and-forward on the Internet, 1998.
12. ITU-T.38: Procedures for Real-time Group 3 Facsimile Communication over IP Networks, Amendment 1, 1999.
13. Handwriting Recognition Group, <hwr.nici.kun.nl>, May, 2008.
14. ITU-T.4: Standardization of Group 3 facsimile terminals for document transmission, 1996

15. ITU-T.30: Procedures for document facsimile transmission in the general switched telephone network, 1996